

CENTRALIZING THE LOCK POINT OF A SYNCHRONOUS CIRCUIT

REFERENCE TO RELATED APPLICATION

5 The disclosure in the present application is related to the disclosure provided in the commonly-assigned United States patent application Serial No. 09/921,614, titled "Method to Improve the Efficiency of Synchronous Mirror Delays and Delay Locked Loops", filed on August 3, 2001.

BACKGROUND

10 Field of the Disclosure

The present disclosure generally relates to synchronous circuits and, more particularly, to a system and method to centralize the lock point of a synchronous circuit.

Brief Description of Related Art

15 Most digital logic implemented on integrated circuits is clocked synchronous sequential logic. In electronic devices such as synchronous dynamic random access memory circuits (SDRAMs), microprocessors, digital signal processors, etc., the processing, storage, and retrieval of information is coordinated or synchronized with a clock signal. The speed and stability of the clock signal determines to a large extent the data rate at which a circuit can function. Many high
20 speed integrated circuit devices, such as SDRAMs, microprocessors, etc., rely upon clock signals to control the flow of commands, data, addresses, etc., into, through and out of the devices.

In SDRAMs or other memory devices, it is desirable to have the data output from the memory synchronized with the system clock that also serves the microprocessor. Delay-locked
25 loops (DLLs) are synchronous circuits used in SDRAMs to synchronize an external clock (e.g., the system clock serving a microprocessor) and an internal clock (e.g., the clock used internally within the SDRAM to perform data read/write operations on various memory cells) with each other. Typically, a DLL is a feedback circuit that operates to feed back a phase difference-related signal to control a delay line, until the timing of one clock signal (e.g., the system clock) is
30 advanced or delayed until its rising edge is coincident (or "locked") with the rising edge of a second clock signal (e.g., the memory internal clock).

Fig. 1 depicts a simplified block diagram of a prior art delay-locked loop (DLL) 10 that can be internal to an SDRAM (not shown). The DLL 10 receives a reference clock 12 as an input and generates an output clock or the CLKOut signal 13 at its output. A Tree_CLK signal 13* is, in turn, fed back as a feedback clock 14 as discussed later. The reference clock 12 is interchangeably referred to herein as “Ref” or “Ref clock signal” or “Ref clock”; whereas the feedback clock 14 is interchangeably referred to herein as “FB” or “FB clock signal” or “FB clock.” The reference clock 12 is typically the external system clock serving the microprocessor or a delayed/buffered version of it. In the embodiment of Fig. 1, the system clock CLK 24 and its inverted version CLK_ 25 are fed into a clock receiver 23 and appear at the receiver’s outputs 26 and 27, respectively. The system clocks are then buffered through a clock buffer 28. One output of the clock buffer 28—i.e., the Ref clock 12—thus is a buffered version of the system clock CLK 24. The other output 30 of the clock buffer 28 is a buffered version of the inverted system clock CLK_ 25 or the inverted version of the Ref clock 12. This output 30 of the clock buffer 28 is interchangeably referred to herein as the “inverted reference clock” or “Ref* clock” or “Ref* clock signal.” In traditional synchronous circuits with a single delay line (e.g., the DLL circuit 10 with a delay line 16), only the Ref clock 12 is input into the delay line as shown in Fig. 1.

The Ref clock 12 may be fed into the delay line 16 via a buffer 15. The output of the buffer 15 is a CLKIn signal 17, which is a buffered version of the reference clock 12. The clock output of the delay line 16—the CLKOut signal 13—is used to provide the internal clock (not shown) used by the SDRAM to perform data read/write operations on memory cells and to transfer the data out of the SDRAM to the data requesting device (e.g., a microprocessor). Thus, as shown in Fig. 1, the CLKOut 13 is sent to a clock distribution network or clock tree circuit 20 whose output 21 may be coupled to SDRAM clock driver and data output stages to clock the data retrieval and transfer operations. As can be seen from Fig. 1, the CLKOut signal 13 (and, hence, the FB clock 14) is generated using a delay line 16, which introduces a specific delay into the input Ref clock 12 (or CLKIn 17) to obtain the “lock” condition.

A phase detector (PD) 18 compares the relative timing of the edges of the system clock CLK 24 and the memory’s internal clock (not shown) by comparing the relative timing of their

respective representative signals—the input clock signal (the Ref clock 12) which relates to the system clock 24, and the FB clock signal 14 which relates to the memory’s internal clock—so as to establish the lock condition. As shown in Fig. 1, an I/O model circuit 22 may be a part of the DLL 10 to function as a buffer or dummy delay circuit for the Tree_CLK signal 13* before the Tree_CLK signal 13* is fed into the phase detector 18 as the FB clock 14. The Tree_CLK signal 13* may be obtained from the clock tree circuit 20 in such a manner as to make the FB clock 14 effectively represent the memory’s internal clock, which may be present through the clock driver and data output stages. The I/O model 22 may be a replica of the system clock receiver 23, the external clock buffer 28, and the clock and data output path (including the clock driver coupled to the output 21) so as to match respective delays imparted by these stages to the system clock 24 and the CLKOut signal 13, thereby making the Ref clock 12 and the FB clock 14 resemble, respectively, the system clock CLK 24 and the internal clock of the memory as closely as possible. Thus, the I/O model 22 attempts to maintain the phase relationship between the Ref clock 12 and the FB clock 14 as close as possible to the phase relationship that exists between the system clock CLK 24 and the memory’s internal clock. The Ref clock 12 and the FB clock 14 are fed as inputs into the phase detector 18 for phase comparison. The output of the PD 18—a delay adjustment signal or indication 19—controls the amount of delay imparted to the CLKIn signal 17 by the delay line 16.

The delay adjustment signal 19 may determine whether the Ref clock 12 should be shifted left (SL) or shifted right (SR) through the appropriate delay in the delay line 16 so as to match the phases of the Ref clock 12 and the FB clock 14 to establish the lock condition. The delay imparted to the Ref clock 12 by the delay line 16 operates to adjust the time difference between the output clock (i.e., the FB clock 14) and the input Ref clock 12 until they are aligned. The phase detector 18 generates the shift left and shift right signals depending on the detected phase difference or timing difference between the Ref clock 12 and the FB clock 14.

Fig. 2 illustrates exemplary timing relationships among various clock signals operated on by the phase detector 18 in the DLL 10 in Fig. 1. The Ref clock 12 and the FB clock 14 are input to the phase detector 18, which generates the shift left or shift right signals depending on whether the rising edge of the Ref clock 12 appears before or after the rising edge of the FB clock 14. In

practice, the DLL 10 is considered “locked” (i.e., the Ref clock 12 and the FB clock 14 are “synchronized”) when the rising edges of the Ref clock 12 and the FB clock 14 are substantially aligned. As shown in part (a) in Fig. 2, when the Ref clock 12 is “leading” or “faster” than the FB clock 14 (i.e., when the rising edge of the Ref clock 12 appears before the rising edge of the FB clock 14) by a time amount equal to “ t_{PE} ”, the PD 18 may generate a shift right (SR) indication to instruct the delay line 16 to right shift the Ref clock 12 by “ t_{PE} ” to achieve the lock condition. Similarly, as shown in part (b) in Fig. 2, when the Ref clock 12 is “slower” than or “lagging” the FB clock 14 (i.e., when the rising edge of the Ref clock 12 appears after the rising edge of the FB clock 14) by the time amount “ t_{PE} ”, the PD 18 may generate a shift left (SL) signal to instruct the delay line 16 to left shift the Ref clock 12 by “ t_{PE} ” to establish the lock. The parameter “ t_{PE} ” ($t_{PE} > 0$) may indicate a small phase error between the FB clock 14 and the Ref clock 12, especially when the FB clock 14 is almost in phase with the Ref clock 12. As discussed below with reference to Fig. 3, in such a situation, the DLL 10 may lock to the either end of the delay line 16.

Fig. 3 depicts delay line lock point locations for the clock signals in Fig. 2 using a traditional DLL locking mechanism (e.g., the DLL 10 in Fig. 1). In conventional DLL locking mechanisms, when the feedback signal (the FB clock 14) is almost in phase with the reference signal (the Ref clock 12), as illustrated in parts (a) and (b) in Fig. 2, the DLL may lock to either end of the delay line 16. Fig. 3 symbolically designates the right end of the delay line 16 as its initial signal entry point 31. In case (b) in Fig. 2, upon entry into the delay line 16 at the initial entry point 31, the Ref clock 12 may be shifted left to establish the lock point 33 that represents a delay of “ t_{PE} ” from the initial entry point 31. Thus, the lock point 33 remains close to the right end of the delay line 16. On the other end, for case (a) in Fig. 2, because the initial entry point 31 is fixed at the right end of the delay line 16, the Ref clock 12 may not be further shifted right by “ t_{PE} ”, but, instead, may have to be shifted left by an amount of delay equal to “ $t_{CK} - t_{PE}$ ” so as to establish the lock point 32 close to the other (left) end of the delay line 16. The clock period of the Ref clock 12 (or the Ref* 30) is designated as “ t_{CK} .” As is observed with reference to the traditional lock point establishment mechanism illustrated in Fig. 3, there may not be enough room for additional tuning or locking range after the initial lock is established or a longer lock time may be required because of the locking performed towards either end of the delay line 16,

instead of towards the center of the delay line. For example, in case of the lock point (b) in Fig. 3, an additional spare delay 34 may be required as part of the delay line 16 for better tuning range (e.g., to accommodate voltage, temperature and frequency fluctuations) after the initial lock 33. The spare delay 34 may increase circuit power consumption and may represent additional hardware. On the other hand, in case of the lock point (a) in Fig. 3, a force-shift-left logic may be required to left shift the Ref clock 12 from the initial entry point 31. Also, in case of lock point (a), longer lock time (from " t_{PE} " to " $t_{CK}-t_{PE}$ ") and, hence, longer delay line 16 may be required.

Therefore, it is desirable to lock a digital synchronous circuit (e.g., a DLL) at the center of or close to the center of the delay line to reduce initial lock time and provide extra tuning range in the event of voltage, temperature and frequency changes after the initial lock is established, but without increasing the size or changing the configuration of the delay line or without requiring a spare delay. When the synchronous circuit is tuned at the center or close to the center of its delay line, more room is available to accommodate voltage, temperature and frequency fluctuations that may affect the initially-established lock.

SUMMARY

The present disclosure contemplates a method of operating a synchronous circuit. The method comprises obtaining a reference clock and an inverted reference clock for the synchronous circuit; using a delay line as part of the synchronous circuit to generate a feedback clock; and selectively using one of the reference clock and the inverted reference clock as an input to the delay line based on a relationship among the phases of the reference clock, the inverted reference clock, and the feedback clock.

In one embodiment, the method of operating a synchronous circuit comprises obtaining a reference clock and an inverted reference clock for the synchronous circuit; using a delay line as part of the synchronous circuit to generate a feedback clock; obtaining a delayed feedback clock from the feedback clock; and selectively using one of the reference clock and the inverted reference clock as an input to the delay line based on a relationship among the phases of the reference clock, the inverted reference clock, the feedback clock and the delayed feedback clock.

In another embodiment, the present disclosure contemplates a method of operating a synchronous circuit, where the method comprises obtaining a reference clock and an inverted reference clock for the synchronous circuit; using a delay line as part of the synchronous circuit to generate a feedback clock; obtaining a delayed feedback clock from the feedback clock; and selectively using one of the reference clock and the inverted reference clock as an input to the delay line based on individual sampling of the reference clock and the inverted reference clock with each of the feedback clock and the delayed feedback clock.

In yet another embodiment, the present disclosure contemplates a method of operating a synchronous circuit, where the method comprises obtaining a reference clock and an inverted reference clock for the synchronous circuit; using a delay line as part of the synchronous circuit to generate a feedback clock; obtaining a delayed feedback clock from the feedback clock; and selectively using one of the reference clock and the inverted reference clock as an input to the delay line based on a relationship among the phases of the reference clock, the inverted reference clock, and one of the feedback clock and the delayed feedback clock.

In a further embodiment, the present disclosure contemplates a synchronous circuit comprising a delay line to receive an input clock and to generate a feedback clock therefrom, wherein the delay line is configured to provide a predetermined delay to the input clock to generate the feedback clock therefrom; and a decoder circuit coupled to the delay line and configured to receive the feedback clock as a first input and to generate a delayed feedback clock therefrom, wherein the decoder circuit is further configured to receive a reference clock as a second input and an inverted reference clock as a third input, wherein the decoder circuit is configured to determine a relationship among the phases of the reference clock, the inverted reference clock, and at least one of the feedback clock and the delayed feedback clock, and to selectively supply one of the reference clock and the inverted reference clock as the input clock to the delay line based on determination of the phase relationship.

In a still further embodiment, the present disclosure contemplates a memory device including a synchronous circuit (e.g., a delay locked loop) constructed according to the teachings

of the present disclosure. In an alternative embodiment, the present disclosure contemplates a system that comprises a processor, a bus, and a memory device coupled to the processor via the bus and including the synchronous circuit.

5 The system and method of the present disclosure establish the lock point of a digital synchronous circuit (e.g., a DLL) at the center of or close to the center of its delay line to provide for extra tuning range in the event of voltage, temperature and frequency changes after the initial lock is established. The selective use of the opposite phase of the reference clock (i.e., the inverted version of the reference clock) for the input of the delay line allows for addition or
10 removal of half cycle of delay to centralize the final lock point of the delay line. The switching between the reference clock and the inverted reference clock results in centralization of the lock point for most cases as well as improvement in the tuning range and the time to establish the initial lock, without requiring an additional delay line or without increasing the size or changing the configuration of the existing delay line.

BRIEF DESCRIPTION OF THE DRAWINGS

For the present disclosure to be easily understood and readily practiced, the present disclosure will now be described for purposes of illustration and not limitation, in connection
20 with the following figures, wherein:

Fig. 1 depicts a simplified block diagram of a prior art delay-locked loop (DLL) that can be internal to an SDRAM;

25 Fig. 2 illustrates exemplary timing relationships among various clock signals operated on by the phase detector in the DLL in Fig. 1;

Fig. 3 depicts delay line lock point locations for the clock signals in Fig. 2 using a traditional DLL locking mechanism;

Fig. 4 is a simplified block diagram of a delay-locked loop (DLL) according to one embodiment of the present disclosure;

Fig. 5 illustrates an exemplary timing relationship among different clock signals operated
5 on by the clock decoder circuit in the DLL in Fig. 4;

Fig. 6 shows delay line lock point locations for the clock signals in Fig. 5 using the DLL locking mechanism of Fig. 4;

10 Fig. 7 illustrates exemplary timing relationship among different clock signals operated on by the phase detector in the DLL in Fig. 4 when the inverted reference clock is input to the delay line by the clock decoder in the DLL;

Fig. 8 depicts an exemplary block diagram showing relevant circuit details according to
15 one embodiment of the present disclosure for the clock decoder in Fig. 4;

Fig. 9 shows an exemplary circuit layout implementing various circuit blocks of the clock decoder depicted in Fig. 8;

20 Fig. 10 illustrates a clock timing relationship that is substantially similar to the timing relationship depicted in part (a) in Fig. 5, except that an additional clock signal—the delayed version of the feedback clock—is also shown;

Fig. 11 depicts another embodiment of the clock decoder shown in Fig. 8;
25

Fig. 12, which is substantially similar to Fig. 9, shows an exemplary circuit layout implementing various circuit blocks of the clock decoder depicted in Fig. 11; and

Fig. 13 is a block diagram depicting a system in which a synchronous circuit (e.g., the
30 DLL in Fig. 4) constructed according to the teachings of the present disclosure may be used.

DETAILED DESCRIPTION

Reference will now be made in detail to some embodiments of the present disclosure, examples of which are illustrated in the accompanying drawings. It is to be understood that the figures and descriptions of the present disclosure included herein illustrate and describe elements that are of particular relevance to the present disclosure, while eliminating, for the sake of clarity, other elements found in typical solid-state memories or memory-based systems. It is noted at the outset that the terms “connected”, “connecting,” “electrically connected,” etc., are used interchangeably herein to generally refer to the condition of being electrically connected. It is further noted that various block diagrams, circuit diagrams and timing waveforms shown and discussed herein employ logic circuits that implement positive logic, i.e., a high value on a signal is treated as a logic “1” whereas a low value is treated as a logic “0.” However, any of the circuit discussed herein may be easily implemented in negative logic (i.e., a high value on a signal is treated as a logic “0” whereas a low value is treated as a logic “1”).

Fig. 4 is a simplified block diagram of a delay-locked loop (DLL) 36 according to one embodiment of the present disclosure. As noted before, the DLL 30 is one type of synchronous circuit that can be internal to any integrated circuit including, for example, an SDRAM memory unit (as shown, for example, in Fig. 13). It is pointed out that throughout the discussion herein the same reference numerals are used to designate identical circuit elements or signal waveforms, and/or to also facilitate ease of discussion. A comparison of the DLL 10 in Fig. 1 and the DLL 36 in Fig. 4 shows that the DLL 36 additionally includes a Clk 180 decoder 38 (interchangeably referred to herein as the “clock decoder” or the “decoder” 180). However, the overall function of the DLL 36 is identical to that of the DLL 10, which is to synchronize the external clock 12 with the internal clock 14.

As shown in Fig. 4, the decoder 38 receives not only the reference clock 12, but also the inverted reference clock 30 as inputs. Thus, even though the DLL 36 is a single delay line DLL, both the reference clocks 12, 30 are input to the DLL 36, instead of just one Ref clock 12 in the DLL 10 in Fig. 1. The decoder 38 also receives the FB clock 14 as an additional input. The clock decoder 38 determines which one of the reference clocks—Ref clock 12 or the Ref* clock 30—should be input to the delay line 16 as the CLKIn signal 17. This determination is made, as

explained later in more detail, based on the phase relationship among the clocks—Ref 12, Ref* 30, and FB 14. In the decoder 38 in Fig. 4, a delayed version of the FB clock (e.g., the FBd clock 43 in Fig. 8) may also be used to determine the phase relationship as discussed later with reference to Figs. 8 and 11. When a certain predetermined phase relationship exists among these four clocks, the decoder 38 supplies the Ref* clock 30 to the delay line 16 instead of the Ref clock 12. It is noted here that the decoding (or clock selection) operation by the decoder 38 is performed prior to any phase comparison by the phase detector 18 and also prior to commencement of any delay adjustment in the delay line 16.

As discussed in more detail later with reference to, for example, Figs. 5-6, the decoder's 38 selection of Ref* 30 instead of Ref 12 to be input to the delay line 16, results in centralization of lock points, especially in situations when the FB clock 14 is almost in phase with the Ref clock 12 (e.g., the timing diagrams (a) and (b) in Fig. 2). Thus, the DLL 36 according to the present disclosure selectively feeds either the Ref clock 12 or the Ref* clock 30 as the clock input (CLKIn 17) to the delay line 16 based on the logic value of a switching signal (SW) 40 (discussed later with reference to Figs. 8-9 and 11-12). The delay line 16 then applies the requisite delay (per signals received from the PD 18) to its input clock CLKIn signal 17, which may be the Ref clock 12 or the Ref* clock 30 depending on the activation of the SW signal 40. As discussed below, the selective use of the opposite phase of the Ref clock 12—i.e., the Ref* clock 30—for the input (the CLKIn signal 17) to the delay line 16 allows for addition or removal of half cycle of delay to centralize the final lock point of the delay line 16. It is noted that the delay line 16 may be a symmetrical delay line, i.e., a delay line that has the same delay whether a high-to-low or a low-to-high logic signal is propagating along the line.

Fig. 5 illustrates exemplary timing relationship among different clock signals operated on by the clock decoder circuit 38 in the DLL 30 in Fig. 4. The Ref clock 12 is shown along with its inverted version Ref* 30. The timing relationships in parts (a) and (b) are identical to the corresponding timing relationships in parts (a) and (b) in Fig. 2, except for the presence of the Ref* clock 30 in Fig. 5. Thus, the timing diagrams in Fig. 5 also illustrate the situation when the feedback signal 14 is almost in phase with the reference signal 12. The minor phase differential (or phase error) is indicated by the time parameter " t_{PE} ." As discussed earlier with reference to

Fig. 3, when the signals (except for the Ref* clock 30) in parts (a) and (b) in Fig. 5 are present in the circuit configuration of Fig. 1, the required delay (shift left) to establish the lock (i.e., aligning the rising edge of Ref 12 with the rising edge of FB 14) is “ $t_{CK}-t_{PE}$ ” and “ t_{PE} ” respectively. However, when Ref* 30 is used (based on the logic level of the SW signal 40 as discussed later with reference to Figs. 8 and 11) instead of Ref 12 as an input CLKIn 17 to the delay line 16 according to the teachings of the present disclosure, the required delay (shift left) to establish the lock may be reduced from “ $t_{CK}-t_{PE}$ ” to “ $\frac{t_{CK}}{2} - t_{PE}$ ” in case (a) in Fig. 5. In case of waveforms in part (b) in Fig. 5, however, the delay (shift left) is increased from “ t_{PE} ” to “ $\frac{t_{CK}}{2} + t_{PE}$ ” when Ref* 30 is used as the CLKIn signal 17. As the delay line 16 may be configured to always shift left initially, when Ref* 30 is used as the CLKIn signal 17, the lock point locations for the waveforms in parts (a) and (b) in Fig. 5 move towards the center of the delay line 16 as discussed with reference to Fig. 6.

Fig. 6 shows delay line lock point locations for the clock signals in Fig. 5 using the DLL locking mechanism of Fig. 4. As can be seen from Figs. 5 and 6, the lock points 32* and 33* for the clocks in parts (a) and (b) in Fig. 5, respectively, are moved substantially to the center of the delay line 16 when Ref* 30 is used as the clock input CLKIn 17 to the delay line 16. Furthermore, because of the centralization of the lock points, the spare delay 34 (Fig. 3) may be removed, thereby reducing the intrinsic delay and power consumption in the delay line 16.

Fig. 7 illustrates exemplary timing relationship among different clock signals operated on by the phase detector 18 in the DLL 36 in Fig. 4 when the inverted reference clock (Ref* 30) is input to the delay line 16 by the clock decoder 38 in the DLL 36. It is observed that the Ref* clock 30 is not input to the PD 18. Instead, only the Ref clock 12 is input to the PD 18 as is the case in the conventional DLL 10. Initially, the Ref clock 12 is input to the delay line 16 as is done in conventional DLL circuits (e.g., the DLL 10 in Fig. 1). Thereafter, the decoder 38 may “observe” the timing relationship between the Ref clock 12 and the FB clock 14, and may determine to input the Ref* clock 30 to the delay line 16 instead of Ref 12 when a predetermined phase relationship (e.g., the waveforms shown in Fig. 5) exists among Ref 12, Ref* 30, and FB

14 as discussed later with reference to Figs. 8 and 11. When the decoder 38 determines to input Ref* 30 to the delay line 16, the waveforms obtained for the FB clock 14 (which is also input to the PD 18) would be half clock cycle ($0.5 t_{CK}$) delayed versions of those waveforms shown in parts (a) and (b) in Fig. 5 as can be observed from a comparison of respective waveforms in Figs. 5 and 7. As can be seen from the exemplary timing diagrams in parts (a) and (b) in Fig. 7, the delay adjustment determined by the PD 18 using the Ref clock 12 and the FB clock 14 (generated from the Ref* clock 30 being input to the delay line 16) is identical to the values given in Fig. 6 for the respective timing waveforms in parts (a) and (b) in Fig. 5. Therefore, even though the reference clocks that are input to the delay line 16 (the Ref* clock 30) and the PD 18 (the Ref clock 12) are different, the computation of delay values remains unaffected.

Fig. 8 depicts an exemplary block diagram showing relevant circuit details according to one embodiment of the present disclosure for the clock decoder 38 in Fig. 4. It is noted here that only those circuit details or circuit elements relevant to the present discussion are shown in Fig. 8. It is understood, however, that the clock decoder 38 in Fig. 8 may include additional circuit elements to make it a fully operational entity as part of, for example, the DLL 36. In the embodiment shown in Fig. 8, the clock decoder 38 receives Ref 12, Ref* 30, and FB 14 as inputs. The delayed feedback clock FBd 43 is generated by inserting a predetermined time delay " t_D " into the FB clock 14 using a delay element 42. The delay " t_D " is internal to the clock decoder 38 and does not affect the delay determinations by the PD 18. The amount of delay " t_D " may determine the location of the lock points and may provide a margin to adjust the locations of the lock points (e.g., closer to right end, or left end, or exact center, etc.) in the delay line 16. It may be desirable to "guard band" t_D to leave the lock point away from the initial entry point. However, as the location of the eventual lock point may not be known beforehand, the predetermined delay " t_D " between FB 14 and FBd 43 may be determined based on how far away the lock point can be moved from the initial entry point, the frequency of the feedback clock 14, and whether the tuning range of the DLL 36 after establishing the initial lock between the reference clock 12 and the feedback clock 14 is enough to accommodate expected PVT (process or frequency, voltage, temperature) variations during run time. For example, in the timing diagram illustrated in Fig. 10, the value of " t_D " is in the range between " t_{PE} " and " $0.5t_{CK}$ " so as to maintain the selection of the Ref* clock 30 as the CLKIn signal 17. If the value of " t_D " does not

fall in this range, then, in the embodiment of Figs. 10-11, for example, the decoder 38 may not select Ref* clock 30 as the CLKIn signal 17 because the switch signal (SW) 40 in Fig. 11 may not be “ON” or “active” in that situation as discussed later below. The delay element 42 may be implemented in a number of ways known in the art including, for example, using AND gates, crossed inverters, an all NAND-based delay element, or a combination of various gate elements (as discussed with reference to Figs. 9 and 12), etc.

As shown in Fig. 8, the Ref clock 12 and the Ref* clock 30 are sampled by the FB clock 14 using a sampler circuit 44 (sampler-1) to determine the phase relationship between Ref 12, Ref* 30, and FB 14. On the other hand, sampler-2 (46) uses the FBd clock 43 to sample the Ref clock 12 and the Ref* clock 30. The sampler circuits 44, 46 may be D-type flip-flops clocked by respective sampling signals FB 14 or FBd 43 whose rising edges sample the reference clocks Ref 12 and Ref* 30. Fig. 9 shows an exemplary circuit layout 60 implementing various circuit blocks of the clock decoder 38 depicted in Fig. 8. As can be seen from Fig. 9, sampler-1 (44) may include the D-type flip-flops 62 and 64 that receive inverted versions (because of the presence of NAND gates 61 and 61*) of the input clocks Ref* 30 (designated as signal DLLR_ in Fig. 9) and Ref 12 (designated as signal DLLR in Fig. 9) respectively. Each input clock (Ref or Ref*) is individually sampled by the FB clock 14 (designated as DLLFB signal in Fig. 9) when the DLLFB signal and its inverted version (generated by the inverter 70 in Fig. 9) are applied as clock inputs to flip-flops 62, 64. Similarly, sampler-2 (46) may include the D-type flip-flops 66 and 68 that also receive the inverted versions of the input clocks Ref 12 (designated as signal DLLR in Fig. 9) and Ref* 30 (designated as signal DLLR_ in Fig. 9) respectively. Each input clock (Ref or Ref*) is individually sampled by the FBd clock 43 (designated as DLLFBd signal in Fig. 9) when the DLLFBd signal and its inverted version (available at the output of the NAND delay element 74 in Fig. 9) are applied as clock inputs to flip-flops 66, 68. It is seen from Fig. 9 that the delay element 42 is implemented through a combination of two NAND delays 72, 74 and an inverter 76. The output of the inverter 76 is the FBd clock 43 in Fig. 8, whereas the output of the NAND delay 74 is the inverted version of FBd to be supplied as a clock input to the D-type flip-flops 66, 68 as noted before. The duration of the delay (“ t_D ”) between FB 14 and FBd 43 clocks may be adjusted by adding another NAND delay 77 or removing one or more of the

NAND delays 72, 74 (and appropriately modifying the circuit connections) into the DLLFB signal in Fig. 9 (i.e., the FB clock 14 in Fig. 8) when needed.

Referring back to Fig. 8, each of the clocks Ref 12 and Ref* 30 is individually sampled with one of the clocks FB 14 and FBd 43 to determine whether a specific phase relationship (e.g., the phase relationships depicted in the timing waveforms in Fig. 5) exists among the clocks Ref 12, Ref* 30, and FB 14. The FBd clock 43 assists in determining the specific phase relationship (discussed later). Each sampler 44, 46 outputs two phase relationship signals—sampler-1 (44) outputting the signals ph0 (48) and ph180 (50), whereas sampler-2 (46) outputting the signals phd0 (54) and phd180 (52). The ph0 signal 48 is generated (i.e., goes “high” or logic “1”) when the sampled value of the reference clock 12 at the time of sampling by the (rising edge of) feedback clock 14 is “high” or logic “1.” On the other hand, the ph180 signal 50 becomes “high” or logic “1” when the sampled value of the inverted reference clock 30 at the time of sampling by the (rising edge of) feedback clock 14 is “high” or logic “1.” Similarly, the phd0 signal 52 is “high” or in logic “1” state when the sampled value of the reference clock 12 at the time of sampling by the (rising edge of) delayed feedback clock 43 is “high” or logic “1”, and the phd180 signal 54 becomes “high” or logic “1” when the sampled value of the inverted reference clock 30 at the time of sampling by the (rising edge of) the delayed feedback clock 43 is “high” or logic “1.” An exemplary generation of these phase relationship signals ph0, ph180, phd0 and phd180 is illustrated in the circuit layout in Fig. 9. In Fig. 9, because the clocks get inverted by the NAND gates 61 and 61*, the ph0 48 and phd0 54 signals are generated using the Ref* clock (the DLLR_ input 30), whereas the ph180 50 and phd180 52 signals are generated using the Ref clock (the DLLR signal 12) to obtain the sampling described hereinbefore. It is observed here that the use of FB 14 and FBd 43 to sample values of Ref 12 and Ref* 30 in samplers 44, 46 results in a determination of phase relationships among Ref, Ref*, FB, and FBd:

- (1) Signals ph0 and ph180 reflect the phase relationship among Ref 12, Ref* 30 and FB 14, and
- (2) signals phd0 and phd180 reflect the phase relationship among Ref 12, Ref* 30, and FBd 43.

As shown in Fig. 8, the outputs ph0, ph180, phd0, and phd180 from the samplers 44, 46 are fed as inputs to a switch signal generator 56, which operates on these inputs according to a predetermined truth table 58 to generate the switching signal SW 40 as its output. In the

embodiment of Fig. 9, the switch signal generator 56 is shown implemented using the logic elements 78-87. From the truth table 58, it is seen that the switching signal 40 is generated or becomes "active" (i.e., logic "high" or "1") under two sets of values (logic 1's or 0's) for the outputs of the samplers 44, 46: (1) When $ph0 = 1$, $ph180 = 0$, $phd0 = 1$, $phd180 = 0$; and (2) when $ph0 = 0$, $ph180 = 1$, $phd0 = 1$, and $phd180 = 0$. These two sets of values may represent whether the FB clock 14 leads or lags the Ref clock 12 by the small phase error " t_{PE} ." These two sets of values may, for example, correspond to the timing relationships in parts (a) and (b), respectively, in Fig. 5 (when appropriate value for " t_D " is set for the generation of the delayed feedback clock 43). In all other combinations of the sampler outputs, the SW signal 40 remains inactive or "low" or logic "0." Thus, the switching signal 40 becomes active only when a specific phase relationship exists among the clocks Ref, Ref*, FB, and FBd, as reflected in the values (given by the truth table 58) of the output signals $ph0$, $ph180$, $phd0$, and $phd180$. That specific phase relationship (indicated by the values in the truth table 58) represents a need to centralize the lock point of the delay line 16 so as to avoid the extreme locking situations discussed hereinbefore with reference to Fig. 3. In all other phase relationships among various input and output clocks, the SW signal 40 is inactive (or "low" or logic "0"), indicating lack of a need to change the input clock phase to centralize the lock point. As noted before, when the SW signal 40 is "ON" or active (logic "1"), the Ref* clock 30 is input into the delay line 16, instead of the Ref clock 12. On the other hand, when the SW signal 40 is "OFF" or inactive (logic "0"), the Ref clock 12 is input into the delay line 16. The switching between the Ref clock 12 and the Ref* clock 30 as the CLKIn input 17 into the delay line 16 allows the DLL 30 to obtain locks that are positioned substantially close to the center of the delay line 16 regardless of whether the FB clock 14 is leading or lagging the input reference clock 12.

It is noted here that the circuit layout 60 in Fig. 9 depicts a portion of the circuit constituting the decoder 38 and is not inclusive of all circuit details of the decoder 38. Thus, as mentioned before, the decoder 38 may include circuits in addition to the circuit blocks shown in Fig. 8 and depicted in the embodiment of Fig. 9 to make the decoder 38 a functioning entity in the DLL 36. As to some other signals present in Fig. 9 (e.g., the system reset signal RST_, the circuit enable signal En, etc.), it is observed that upon the system reset (using the RST_ signal), the enable signal (En) is generated and the operation of the decoder 38 may be activated by the

combination of the enable DLL (ENDLL) and DLL_PD180 signals in Fig. 9. As mentioned before, the decoding or phase comparison operation (by the decoder 38) is carried out prior to the commencement of phase detection by the PD 18. After the phase relationship determination is made by the decoder 38 and after the appropriate reference clock (Ref 12 or Ref* 30) is input to the delay line 16, the decoder 38 may be deactivated. Thereafter, the PD 18 may commence the delay adjustment operations in the conventional manner (as described before with reference to Fig. 1). There may be no real-time phase relationship determination by the decoder 38. The relevant clock input (either Ref 12 or Ref* 30) to the delay line 16 may continue until further activation of the decoder 38 to determine whether a change in the clock input (based on change in existing phase relationship) is necessary to maintain centralized lock points.

Fig. 10 illustrates a clock timing relationship that is substantially similar to the timing relationship depicted in part (a) in Fig. 5, except that an additional clock signal—the delayed version of the feedback clock 14 (i.e., the FBd signal 43)—is also shown. As in Fig. 8, the FBd signal 43 is also used in the embodiment of Fig. 10 (and Figs. 11-12) to check whether the FB clock 14 is close to the Ref clock 12 or the Ref* clock 30 so as to move the lock point away from the initial delay line entry point. In the exemplary timing waveforms shown in Fig. 10, it is observed that the rising edge of the FBd signal 43 is between the rising edges of the Ref clock 12 and the Ref* clock 30. In that situation, as discussed below with reference to Fig. 11, the decoder 38 may be configured to always switch the phase of the CLKIn input 17 from the Ref clock 12 to the Ref* clock 30. It is observed, however, that the decoder 38 may perform the switching (and, hence, centralize the lock point) so long as the amount of delay “ t_D ” is given by: $t_{PE} < t_D < \frac{t_{CK}}{2}$.

Hence, the value of “ t_D ” may guardband how far the lock point may move from the initial entry point. For example, if value of “ t_D ” becomes greater than $0.5t_{CK}$, then the relationship between phd0 signal 54 and the phd180 signal 52 in the truth table 58 in Fig. 11 (discussed below) may not hold valid. In that case, the decoder 38 may not generate the switch signal 40 and, hence, the Ref clock 12 may continue to be the CLKIn input 17. When that occurs, the phase detector 18 may apply the delay adjustment equal to “ $t_{CK}-t_{PE}$ ” because the timing relationship between the Ref clock 12 and the FB clock 14 would be similar to that discussed hereinbefore with reference to part (a) in Fig. 2 (having the corresponding lock point location 32 in Fig. 3) and will not be

similar to that discussed with reference to the waveforms in part(a) in Fig. 7. Hence, proper determination of “ t_D ” may be necessary to achieve lock point centralization.

Fig. 11 depicts another embodiment of the clock decoder 38 shown in Fig. 8. It is noted here that the same reference numerals are used to refer to similar circuit elements in Figs. 8 and 11 for the sake of clarity of discussion and ease of comparison between the embodiments in Figs. 8 and 11. It is evident to one skilled in the art that the block diagram of the decoder in Fig. 11 is quite different (less complex) from that for the decoder in Fig. 8. Thus, although the same reference numeral “38” is used to refer to both the decoders in Figs. 8 and 11 for ease of discussion, it is observed here that Figs. 8 and 11 represent block diagrams of two entirely different embodiments of the decoder 38 in Fig. 4. The similar observations also apply to, for example, the switch signal generator 56 in Figs. 8 and 11.

A comparison of the block diagrams of decoders in Figs. 8 and 11 indicates two major differences between the decoder in Fig. 8 and that in Fig. 11: (1) The decoder 38 in Fig. 11 does not contain the sampler-1 unit 44, i.e., the decoder in Fig. 11 does not use the FB clock 14 directly to sample the Ref 12 and Ref* 30 clocks, but, instead, uses only the FBd clock 43 (which is, of course, related in phase with the FB clock 14) to establish the phase relationship among Ref 12, Ref* 30 and FBd 43; and (2) the truth table 58 for the decoder in Fig. 11 lists only a single set of values of the phd0 signal 54 and the phd180 signal 52 to determine when the switching signal (SW) 40 is generated. Thus, the decoder 38 in Fig. 11 indirectly determines the relationship among the phases of Ref 12, Ref* 30, and FB 14 using the FBd clock 43 instead of the FB clock 14 for sampling. However, because the overall function (and many circuit elements) of the decoder in Fig. 11 is identical to that of the decoder in Fig. 8—i.e., to supply the Ref* clock 30 as the CLKIn signal 17 when a specific phase relationship among Ref 12, Ref* 30, FB 14 and FBd 43 exists, a detailed description of the operation of the decoder 38 in Fig. 11 is not provided. It is seen that the values of phd0 (logic “1”) and phd180 (logic “0”) given in the truth table 58 in Fig. 11 may be obtained when the rising edge of the FBd clock 43 is between the rising edges of the Ref clock 12 and the Ref* clock 30, in that order, as depicted, for example, in the waveforms in Fig. 10. In all other situations—for example, when the rising edge of the FBd clock 43 occurs after the rising edge of the Ref* clock 30, but before the rising edge

of the Ref clock 12—the values of phd0 and phd180 may not be those required by the truth table 58 in Fig. 11 to generate the switching signal (SW) 40 and, hence, the decoder 38 may not supply the Ref* clock 30 as the CLKIn input 17. Thus, when the decoder 38 in Fig. 11 selects the Ref* clock 30 as the CLKIn input 17, the delay to establish the lock point for the timing relationship in Fig. 10 will be reduced from “ $t_{CK-t_{PE}}$ ” to “ $0.5t_{CK-t_{PE}}$ ” as discussed hereinbefore with reference to the similar timing diagram in part (a) in Fig. 5.

Fig. 12, which is substantially similar to Fig. 9, shows an exemplary circuit layout 120 implementing various circuit blocks of the clock decoder 38 depicted in Fig. 11. Based on a comparison of the circuit layout 60 in Fig. 9 and the layout 120 in Fig. 12, it is seen that: (1) The sampler-1 (44) is not implemented in Fig. 12, but only the sampler-2 (46) is implemented; (2) the switch signal generator 56 in Fig. 11 is implemented using the logic elements 124, 125, and 87 in Fig. 12, as opposed to a more extensive logic network of elements 78-87 in Fig. 9; and (3) the delay element 42 in Fig. 11 is implemented in Fig. 12 using the two NAND delays 72, 74 and three inverters 76, 121, and 122 (as opposed to just one inverter 76 in Fig. 9). In Fig. 12, the switching signal (SW) 40 becomes “ON” or “active” (logic “1”) when the phd0 signal 54 is logic “1” and the phd180 signal 52 is logic “0.” As explained with reference to Fig. 11, for all other values of the phd0 and phd180 signals, the SW signal 40 in Fig. 12 remains “OFF” (logic “0”). The rest of the circuit layout in Fig. 12 is similar to that in Fig. 9 and, hence, no additional discussion of the circuit layout 120 in Fig. 12 is provided.

Although the discussion given hereinbefore is with reference to a DLL, the delay line lock point centralization methodology of the present disclosure may be used with any other synchronous circuit including, for example, a synchronous mirror delay circuit (SMD) that may also be used for clock synchronization in various electronic integrated circuits including, for example, SDRAMs.

It is known that the locking range of a delay line is determined by the number of delay stages in the delay line. In one embodiment, the methodology of the present disclosure may be used along with the methodology in the co-pending, commonly-assigned United States patent application titled “A Method to Improve the Efficiency of Synchronous Mirror Delays and Delay

Locked Loops”, which was filed on August 31, 2001 and describes how to reduce the number of delay line stages without reducing the locking range of the delay line. In an alternative embodiment, the teachings in the patent application mentioned in the previous sentence may be used to achieve centralization of a lock point in a manner similar to that discussed hereinbefore with reference to the present disclosure. That patent application describes another method of determining when and whether to use a clock (similar to Ref 12) or its inverted version (similar to Ref* 30) as an input clock to a synchronous circuit (e.g., an SMD circuit).

It is observed that the duty cycle distortion of the reference clock Ref 12 (for example, when the Ref clock 12 is not 50% duty cycle) may cause problems when the inverted reference clock Ref* 30 is used as the CLKIn signal 17. In an embodiment with a single delay line (e.g., the delay line 16 in Fig. 4), a duty cycle corrector (analog or digital correction) may be used to recover the duty information as is known in the art. In an alternative embodiment, two separate delay lines (not shown) may be used in the DLL 36 or the delay line 16 may constitute two separate delay lines (not shown). One of the two delay lines may be for the Ref clock 12 only, whereas the other delay line may be for the Ref* clock 30 only. At any given time, both the delay lines may be active, however, the selection of the clock inputs (Ref 12 or Ref* 30) of the delay lines may be controlled using the SW signal 40. Using two separate delay lines—one each for the Ref 12 and Ref* 30 clocks—may maintain the duty cycle information between the Ref 12 and Ref* 30 clocks in the same relation to each other. If either Ref 12 or Ref* 30 has duty cycle distortion, but their relative timing is still $0.5t_{CK}$, then the dual delay line with proper matching can maintain the relative timing. The delay line input clock selection methodology of the present disclosure may be well suited for a dual delay line configuration.

Fig. 13 is a block diagram depicting a system 90 in which a synchronous circuit (e.g., the DLL 36 in Fig. 4) constructed according to the teachings of the present disclosure may be used. The system 90 may include a data processing unit or computing unit 91 that includes a processor 92 for performing various computing functions, such as executing specific software to perform specific calculations or data processing tasks. The computing unit 91 may also include a memory device 94 that is in communication with the processor 92 through a bus 96. The bus 96 may include an address bus (not shown), a data bus (not shown), and a control bus (not shown). The

memory device 94 can be a dynamic random access memory (DRAM) chip or another type of memory circuits such as SRAM (Static Random Access Memory) chip or Flash memory. Furthermore, the DRAM could be a synchronous DRAM commonly referred to as SGRAM (Synchronous Graphics Random Access Memory), SDRAM (Synchronous Dynamic Random Access Memory), SDRAM II, or DDR SDRAM (Double Data Rate SDRAM), as well as Synchlink or Rambus DRAMs. Those of ordinary skill in the art will readily recognize that memory device 94 of Fig. 13 is simplified to illustrate one embodiment of a memory device and is not intended to be a detailed illustration of all of the features of a typical memory chip. The processor 92 can perform a plurality of functions based on information and data stored in the memory device 94. The processor 92 can be a microprocessor, digital signal processor, embedded processor, micro-controller, dedicated memory test chip, or the like.

The memory device 94 may include a plurality of memory cells 98 generally arranged in rows and columns to store data in rows and columns. A row decode circuit 100 and a column decode circuit 102 may select the rows and columns in the memory cells 98 in response to decoding an address, provided on the bus 96. Data to/from the memory cells 98 is then transferred over the bus 96 via sense amplifiers and a data output path (not shown). A memory controller 104 controls data communication to and from the memory device 94 via an I/O (input/output) circuit 106 in response to control signals (not shown) on the bus 96. The I/O circuit 106 may include a number of data output buffers to receive the data bits from the memory cells 98 and provide those data bits or data signals to the corresponding data lines in the bus 96. The memory controller 104 may include a command decode circuit (not shown). The command decode circuit may receive the input control signals (on the bus 96) (not shown) to determine the modes of operation of the memory device 94. Some examples of the input signals or control signals (not shown in Fig. 8) on the bus 96 include an External Clock signal, a Chip Select signal, a Row Access Strobe signal, a Column Access Strobe signal, a Write Enable signal, etc.

The memory device 94 may also include the DLL 36 according to one embodiment of the present disclosure and a strobe generation circuit 108. The DLL circuit 36 may provide a clock signal (e.g., the FB clock 14) to output a data signal, which is read from memory cells 98 during a data read operation. The strobe generation circuit 108 may provide a strobe signal that may be

sent to the DLL 36 that, in turn, may introduce a delay into the strobe signal so as to synchronize the strobe signal with the data signal during the data read operation from memory cells 98. The strobe signal is a reference signal or timing signal, which tracks the data signal when the data is output. Other circuits or devices in the computing unit 91, based on the timing of the strobe
5 signal, can accurately capture or “latch” the data signal.

The system 90 may include one or more input devices 110 (e.g., a keyboard or a mouse) connected to the computing unit 91 to allow a user to manually input data, instructions, etc., to operate the computing unit 91. One or more output devices 112 connected to the computing unit
10 91 may also be provided as part of the system 90 to display or otherwise output data generated by the processor 92. Examples of output devices 112 include printers, video terminals or video display units (VDUs). In one embodiment, the system 90 also includes one or more data storage devices 114 connected to the data processing unit 91 to allow the processor 92 to store data in or retrieve data from internal or external storage media (not shown). Examples of typical data
15 storage devices 114 include drives that accept hard and floppy disks, CD-ROMs (compact disk read-only memories), and tape cassettes.

The foregoing describes a system and method to establish the lock point of a digital synchronous circuit (e.g., a DLL) at the center of or close to the center of its delay line to provide
20 for extra tuning range in the event of voltage, temperature and frequency changes after the initial lock is established. The delay line receives a clock signal as its input and imparts a given delay to the clock signal to generate a feedback clock that is synchronized or “locked” with the clock signal. The synchronous circuit is configured to selectively use either a reference clock or its inverted version (an inverted reference clock) as the clock signal input to the delay line based on
25 a relationship among the phases of the reference clock, the inverted reference clock, and the feedback clock. A delayed version of the feedback clock may be used during determination of the phase relationship. The selective use of the opposite phase of the reference clock for the input of the delay line allows for addition or removal of half cycle of delay to centralize the final lock point of the delay line. The switching between the reference clock and the inverted reference
30 clock results in centralization of the lock point for most cases as well as improvement in the

tuning range and the time to establish the initial lock, without requiring an additional delay line or without increasing the size or changing the configuration of the existing delay line.

While the disclosure has been described in detail and with reference to specific
5 embodiments thereof, it will be apparent to one skilled in the art that various changes and modifications can be made therein without departing from the spirit and scope of the embodiments. Thus, it is intended that the present disclosure cover the modifications and variations of this disclosure provided they come within the scope of the appended claims and their equivalents.

10